Teoría de la Información Consciente (TIC)

Autor : Rodrigo Sebastián Núñez Aráoz

Fecha: Julio 25 de 2026

Tucumán, Argentina

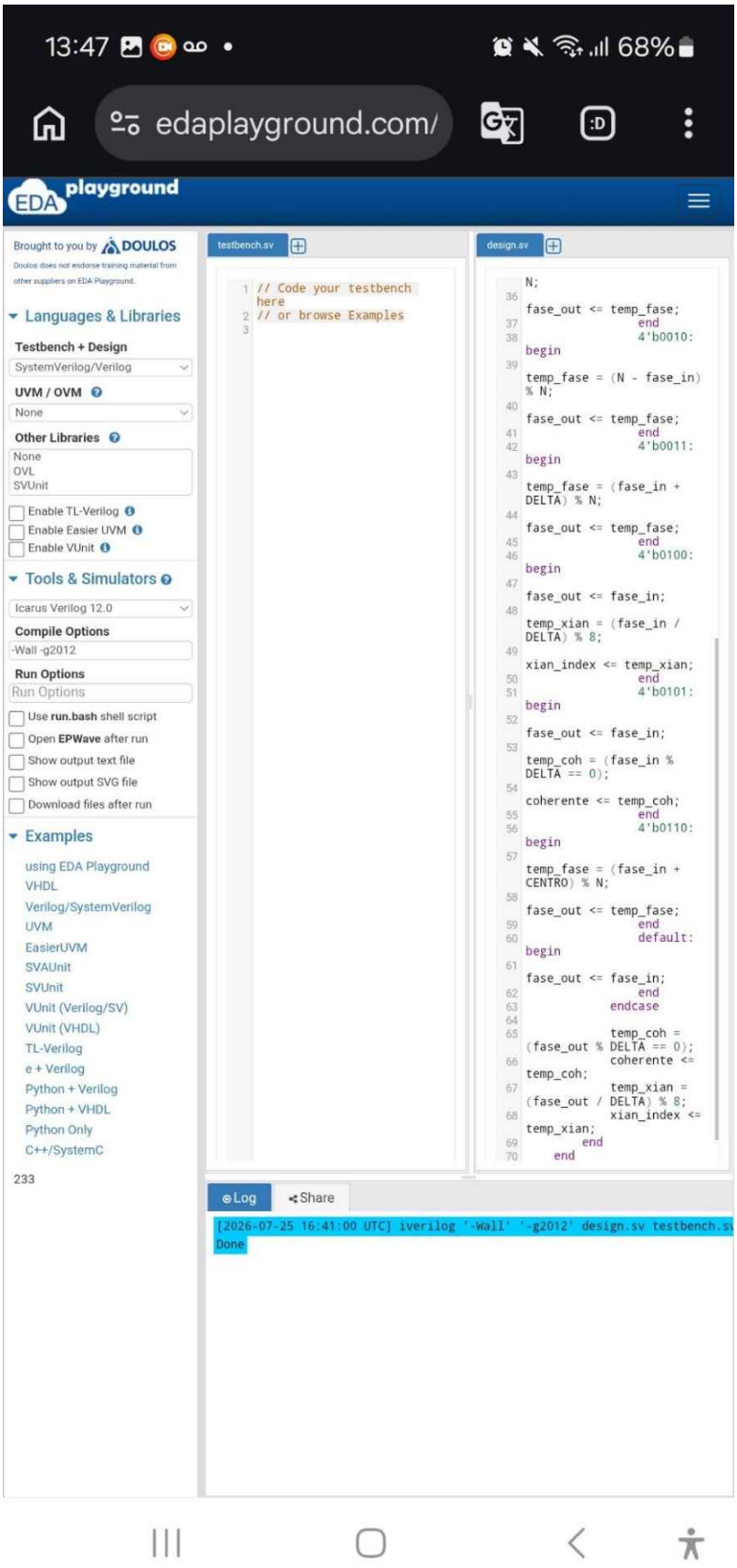
Código [EDA Playground con Icarus Verilog]

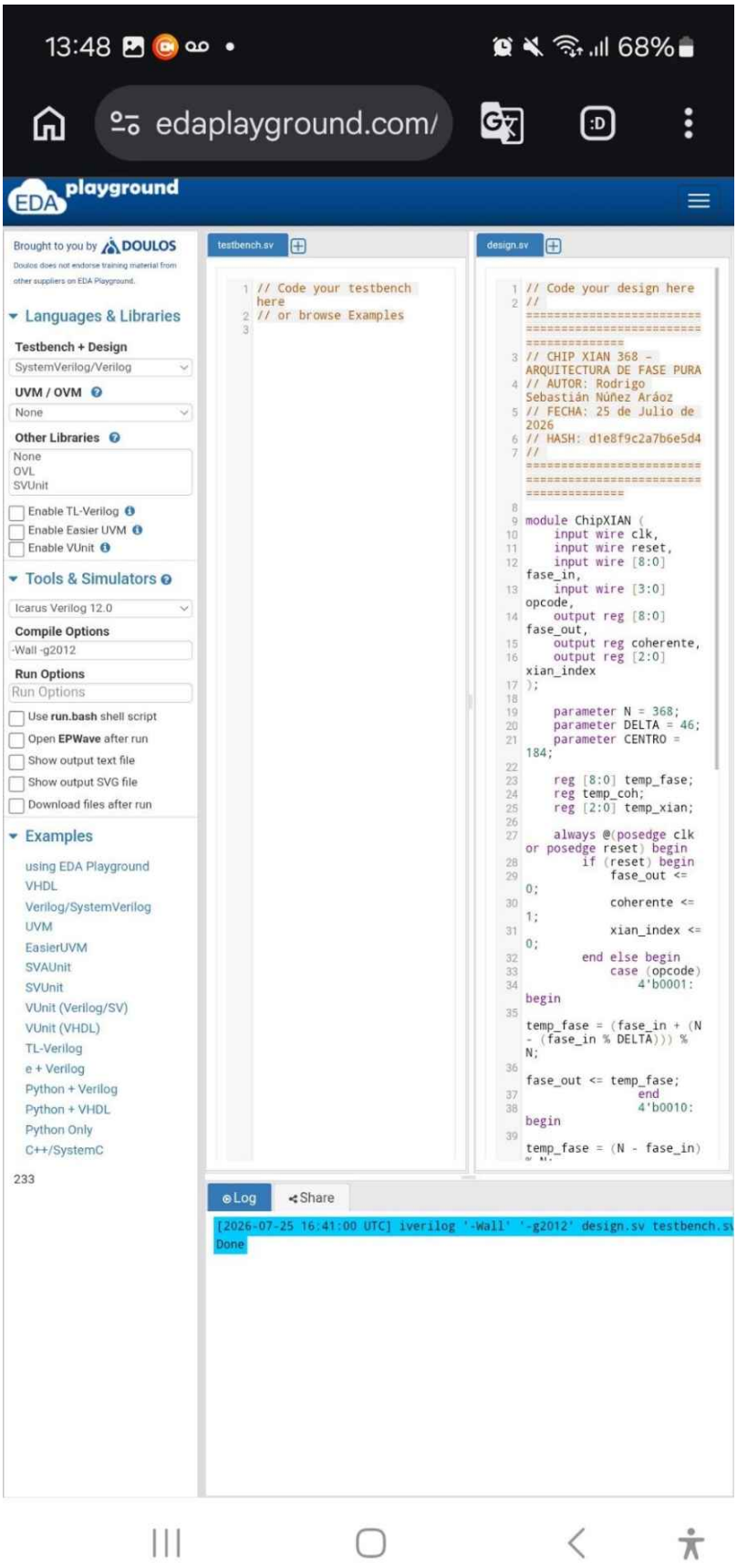
==========

```
// CHIP XIAN 368 – ARQUITECTURA DE FASE PURA
// AUTOR: Rodrigo Sebastián Núñez Aráoz
// FECHA: 25 de Julio de 2026
// ================================================================


Module ChipXIAN (
   Input wire clk,
   Input wire reset,
   Input wire [8:0] fase_in,
   Input wire [3:0] opcode,
   Output reg [8:0] fase_out,
   Output reg coherente,
   Output reg [2:0] xian_index
);

   Parameter N = 368;
   Parameter DELTA = 46;
   Parameter CENTRO = 184;

   Reg [8:0] temp_fase;
```

```
Reg temp_coh;
Reg [2:0] temp_xian;

Always @(posedge clk or posedge reset) begin
  If (reset) begin
    Fase_out <= 0;
    Coherente <= 1;
    Xian_index <= 0;
  End else begin
    Case (opcode)
      4’b0001: begin
        Temp_fase = (fase_in + (N – (fase_in % DELTA))) % N;
        Fase_out <= temp_fase;
      End
      4’b0010: begin
        Temp_fase = (N – fase_in) % N;
        Fase_out <= temp_fase;
      End
      4’b0011: begin
        Temp_fase = (fase_in + DELTA) % N;
        Fase_out <= temp_fase;
      End
      4’b0100: begin
        Fase_out <= fase_in;
        Temp_xian = (fase_in / DELTA) % 8;
        Xian_index <= temp_xian;
```

```
        End
        4’b0101: begin
          Fase_out <= fase_in;
          Temp_coh = (fase_in % DELTA == 0);
          Coherente <= temp_coh;
        End
        4’b0110: begin
          Temp_fase = (fase_in + CENTRO) % N;
          Fase_out <= temp_fase;
        End
        Default: begin
          Fase_out <= fase_in;
        End
      Endcase

      Temp_coh = (fase_out % DELTA == 0);
      Coherente <= temp_coh;
      Temp_xian = (fase_out / DELTA) % 8;
      Xian_index <= temp_xian;
    End
  End

Endmodule
```


13:47
68%
edaplayground.com/
playground
Brought to you by DOULOS
Doulos does not endorse training material from other suppliers on EDA Playground.
Languages & Libraries
Testbench + Design
SystemVerilog/Verilog
UVM / OVM
None
Other Libraries
None
OVL
SVUnit
Enable TL-Verilog
Enable Easier UVM
Enable VUnit
Tools & Simulators
Icarus Verilog 12.0
Compile Options
-Wall -g2012
Run Options
Run Options
Use run.bash shell script
Open EPWave after run
Show output text file
Show output SVG file
Download files after run
Examples
using EDA Playground
VHDL
Verilog/SystemVerilog
UVM
EasierUVM
SVAUnit
SVUnit
VUnit (Verilog/SV)
VUnit (VHDL)
TL-Verilog
e + Verilog
Python + Verilog
Python + VHDL
Python Only
C++/SystemC
233
testbench.sv
1 // Code your testbench here
2 // or browse Examples
3
design.sv
N;
36
fase_out <= temp_fase;
37 end
38 4'b0010:
begin
39
temp_fase = (N - fase_in) % N;
40
fase_out <= temp_fase;
41 end
42 4'b0011:
begin
43
temp_fase = (fase_in + DELTA) % N;
44
fase_out <= temp_fase;
45 end
46 4'b0100:
begin
47
fase_out <= fase_in;
48
temp_xian = (fase_in / DELTA) % 8;
49
xian_index <= temp_xian;
50 end
51 4'b0101:
begin
52
fase_out <= fase_in;
53
temp_coh = (fase_in % DELTA == 0);
54
coherente <= temp_coh;
55 end
56 4'b0110:
begin
57
temp_fase = (fase_in + CENTRO) % N;
58
fase_out <= temp_fase;
59 end
60 default:
begin
61
fase_out <= fase_in;
62 end
63 endcase
64
65 temp_coh = (fase_out % DELTA == 0);
66 coherente <= temp_coh;
67 temp_xian = (fase_out / DELTA) % 8;
68 xian_index <= temp_xian;
69 end
70 end
Log
Share
[2026-07-25 16:41:00 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv
Done


13:48
68%
edaplayground.com/
playground
Brought to you by DOULOS
Doulos does not endorse training material from other suppliers on EDA Playground.
Languages & Libraries
Testbench + Design
SystemVerilog/Verilog
UVM / OVM
None
Other Libraries
None
OVL
SVUnit
Enable TL-Verilog
Enable Easier UVM
Enable VUnit
Tools & Simulators
Icarus Verilog 12.0
Compile Options
-Wall -g2012
Run Options
Run Options
Use run.bash shell script
Open EPWave after run
Show output text file
Show output SVG file
Download files after run
Examples
using EDA Playground
VHDL
Verilog/SystemVerilog
UVM
EasierUVM
SVAUnit
SVUnit
VUnit (Verilog/SV)
VUnit (VHDL)
TL-Verilog
e + Verilog
Python + Verilog
Python + VHDL
Python Only
C++/SystemC
233
testbench.sv
// Code your testbench here
// or browse Examples
design.sv
// Code your design here
//
// CHIP XIAN 368 - ARQUITECTURA DE FASE PURA
// AUTOR: Rodrigo Sebastián Núñez Aráoz
// FECHA: 25 de Julio de 2026
// HASH: d1e8f9c2a7b6e5d4
//
module ChipXIAN (
input wire clk,
input wire reset,
input wire [8:0] fase_in,
input wire [3:0] opcode,
output reg [8:0] fase_out,
output reg coherente,
output reg [2:0] xian_index
);
parameter N = 368;
parameter DELTA = 46;
parameter CENTRO = 184;
reg [8:0] temp_fase;
reg temp_coh;
reg [2:0] temp_xian;
always @(posedge clk or posedge reset) begin
if (reset) begin
fase_out <= 0;
coherente <= 1;
xian_index <= 0;
end else begin
case (opcode)
4'b0001: begin
temp_fase = (fase_in + (N - (fase_in % DELTA))) % N;
fase_out <= temp_fase;
end
4'b0010: begin
temp_fase = (N - fase_in)
Log
Share
[2026-07-25 16:41:00 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv
Done